The Land PEATE VIIRS Library Setup for Code under the NPP2 Repository

This document assumes you have obtained a tagged version of a PGE, and therefore have all of the Library and Config files needed.

1. To set up so that the machine can find all the requisite HDF libraries, all the include files used, and the dynamic libraries used, you need to do two things:
   - set an environment variable named DEV to the directory where your tagged code resides, e.g. “setenv DEV /home/me/<tagged name>” (example of a tagged directory: /home/cdavidson/CM_OPS_PGE353_VP1.5.0)
   - source $DEV/Config/dev_NPP_OPS_env_setup *or* $DEV/Config/dev_NPP_OPS_env_setup_debug; the latter setup sets all flags for the compiler to include debugging flags. Similarly to set up to use the SCIENCE part of the library, source $DEV/dev_NPP_SCI_env_setup *or* $DEV/dev_NPP_SCI_env_setup_debug

2. A function library in the OPS section has been built which should make almost all actions that programmers have to do straightforward with the details hidden “under the hood”. The library is written in C++. Its pieces are:
   - A universal listing of all the parameters which might be used by PGEs and code which reads all the parameters.
   - An HDF (4 for now) library which reads, creates, and writes to HDF files.
   - A set of controls which automatically decide how to step through input files based on how the OPS code itself is set up (this changes from time to time but should be transparent to the user).
   - Functions which allow transfer of data from HDF files to memory structures. You as a programmer should not have to worry about allocation of space; it will be done automatically. Likewise freeing space.
   - A listing of swath names (as an include file) and a listing of field names (also as an include file) which are to be used throughout the code.
   - Parts of the IDPS Libraries which are used by some or all of the PGEs.

3. Compiling the libraries: The precompiled production versions of the libraries are under the <tagged directory>/lib32 and <tagged directory>/lib64 directories and should not require recompiling. Two instances where compiling the libraries would be needed are if you modify the library code or you need the debug variation of the libraries. If recompiling is necessary, go to the <tagged directory>/Library/buildfiles area. Here you will find multiple build scripts named like “dev_buildlibs_....” to aid in compiling the assorted libraries. Each build contains an informative header indicating optional arguments. There is also a README.txt file in this directory that explains which files and libraries go together.

See the document: OPS_CODE_CHECKLIST.pdf for additional information on the library and directory structures and configuring your environment.

4. Once you have your libraries in place and you have sourced the environment setup script, you are ready to compile your code. There should be a README.txt file in the PGE COMB subdirectory ($DEV/OPS/PGExxx/COMB) that provides direction on compiling the PGE.